SG-Net: Syntax-Guided Machine Reading Comprehension

Zhuosheng Zhang[1,2,3,*], **Yuwei Wu**[1,2,3,4,*], **Junru Zhou**[1,2,3], **Sufeng Duan**[1,2,3], **Hai Zhao**[1,2,3,†]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China
[3]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai, China
[4]College of Zhiyuan, Shanghai Jiao Tong University, China
{zhangzs,will8821,zhoujunru,1140339019dsf}@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn

Abstract

For machine reading comprehension, how to effectively model the linguistic knowledge from the detail-riddled and lengthy passages and get ride of the noises is essential to improve its performance. In this work, we propose using syntax to guide the text modeling of both passages and questions by incorporating syntactic clues into multi-head attention mechanism to fully fuse information from both global and attended representations. Accordingly, we present a novel syntax-guided network (SG-Net) for challenging reading comprehension tasks. Extensive experiments on popular benchmarks including SQuAD 2.0 and RACE validate the effectiveness of the proposed method with substantial improvements over fine-tuned BERT. This work empirically discloses the effectiveness of syntactic structural information for text modeling. The proposed attention mechanism also verifies the practicability of using linguistic information to guide attention learning and can be easily adapted with other tree-structured annotations.

Introduction

Recently, much progress has been made in general-purpose language modeling that can be used across a wide range of tasks (Peters et al. 2018; Radford et al. 2018; Devlin et al. 2018). Understanding the meaning of a sentence is a prerequisite to solve many natural language understanding (NLU) problems, such as reading comprehension based question answering (Rajpurkar, Jia, and Liang 2018). Obviously, it requires a good representation of the meaning of a sentence.

Human reads most words superficially and pays more attention to the key ones during reading and understanding sentences (Wang, Zhang, and Zong 2017), that is inconsistent with the way in which most of the existing models do. These models equally tackle each word in a sentence without distillation. Specifically, if the text is particularly lengthy and detailed-riddled, it would be quite difficult for a deep learning based model to understand as it suffers from noise and pays vague attention on the text components (Mudrakarta et al. 2018), let alone accurately answering questions. In contrast, extensive studies have verified that human reads sentences efficiently by taking a sequence of fixation and saccades and prefers to re-read the long texts such as passages after a quick first glance (Yu, Lee, and Le 2017).

Though a sort of attention mechanisms tried to weigh out the important parts of an input sequence, they still do calculations for each word without explicit pruning and prior focus. For passage involved reading comprehension, the input sequence always consists of multiple sentences. Nearly all of the current attention methods regard the sentences as a whole, e.g. a passage, with little consideration of the inner linguistic structure inside each sentence. This would result in process bias caused by much noise and lack of associated spans for each concerned word. All these factors motivate us to seek for an informative method that can selectively pick out important words and linguistically distinguish every word, instead of taking all the words, only considering the related subset of words of syntactic importance inside each input sentence explicitly with a guidance of syntactic structure clues to give more accurate attentive signals and reduce the impact of noise brought about by the whole passage.

In this paper, we introduce syntax to guide machine reading comprehension by building a fine-grained structured representation for each sentence to improve language modeling and further propose a syntax-guided network for machine reading comprehension. Specifically, we adopt pre-trained dependency syntactic parse tree structure to produce the related nodes for each word in a sentence, namely dependency of interest (DOI), by regarding each word as a child node and the DOI is the set of its parent nodes in the dependency parsing tree. An example is shown in Figure 1. Then we propose syntax-guided self attention to incorporate the syntactic tree structure information and explore ways of aggregation to form syntax-enhanced representations. Our evaluations are based on two widely used challenging MRC tasks, span-based SQuAD 2.0 and multi-choice style RACE.

To our best knowledge, we are the first to integrate syntactic relationship as attentive guidance for machine reading comprehension and propose a general syntax-guided structure for deep aggregation. A series of experiments and analysis show the proposed method is effective and boosts the strong BERT baseline substantially.

[*]These authors contribute equally. † Corresponding author.

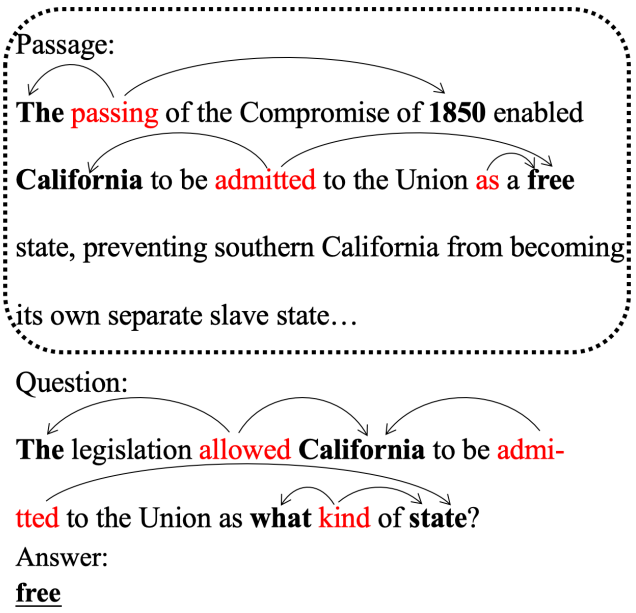


Figure 1: Example of span-based QA. Our proposed model is capable of capturing syntactic information provided by sentence parser, which is very helpful for predicting the final answer. The main head words in the sentences are shown in red and words between left and right arrows are child nodes of the head word. The DOI of each word is a set of its head (parent) words.

Related Work

Machine Reading Comprehension

In the last decade, the MRC tasks have evolved from the early cloze-style test (Hill et al. 2015; Hermann et al. 2015; Zhang, Huang, and Zhao 2018) to span-based answer extraction from passage (Rajpurkar et al. 2016; Nguyen et al. 2016; Joshi et al. 2017; Wang et al. 2018; Rajpurkar, Jia, and Liang 2018) and multi-choice style ones (Lai et al. 2017) where the two latter ones are our focus in this work. A wide range of attentive models have been employed, including Attention Sum Reader (Kadlec et al. 2016), Gated attention Reader (Dhingra et al. 2017), Self-matching Network (Wang et al. 2017), Attention over Attention Reader (Cui et al. 2017) and Bi-attention Network (Seo et al. 2016). Meanwhile, researchers are also investigating models with more complex context understanding(Wang, Yan, and Wu 2018; Wang et al. 2018; Zhang and Zhao 2018; Zhang et al. 2018).

Recently, deep contextual language model has shown effective for learning universal language representations by leveraging large amounts of unlabeled data, achieving various state-of-the-art results in a series of NLU benchmarks. Some prominent examples are Embedding from Language models (ELMo) (Peters et al. 2018), Generative Pre-trained Transformer (OpenAI GPT) (Radford et al. 2018) and Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al. 2018) among which BERT uses a different pre-training objective, masked language model, which allows capturing both sides of context, left and right. Besides, BERT also introduces a *next sentence prediction* task that jointly pre-trains text-pair representations. The latest evaluation shows that BERT is powerful and convenient for downstream tasks, which could be either easily applied to downstream models as the encoder or directly used for fine-tuning. Following this line, we extract context-sensitive syntactic features and take pre-trained BERT as our backbone encoder for empowering explicit syntactic dependencies.

Syntactic Structures

Recently, dependency syntactic parsing have been further developed with neural network and attained new state-of-the-art results (Chen and Manning 2014; Dozat and Manning 2016; Ma et al. 2018). Benefiting from the highly accurate parser, neural network models could enjoy even higher accuracy gains by leveraging syntactic in-formation rather than ignoring it (Roth and Lapata 2016; He et al. 2017; Marcheggiani and Titov 2017).

Syntactic dependency parse tree provides a form that is capable of indicating the existence and type of linguistic dependency relation among words, which has been shown generally beneficial in various natural language understanding tasks (Bowman et al. 2016). To effectively exploit syntactic clue, most of previous works absorb parse tree information by transforming dependency labels into vectors and simply concatenate the label embedding with word representation. However, such simplified and straightforward processing would result in higher dimension of joint word and label embeddings and is too coarse to capture contextual interactions between the associated labels and the mutual connections between labels and words. This inspires us to seek for an attentive way to enrich the contextual representation from the syntactic source. A similar work is from (Strubell et al. 2018), which proposed to incorporate syntax with multi-task learning for semantic role labeling. However, their syntax is incorporated by training one extra attention head to attend to syntactic parents for each token. Instead of only considering syntactic head nodes, we propose to fully use all the DOI parts and we extend the syntax-guided representation to conjunct with original contextual word embedding via a bi-attention. Thus, we form a general approach to benefit from syntax-guided representations, which is the first attempt for machine reading comprehension to our best knowledge.

Syntax-Guided Network

Our goal is to design an effective neural network model which makes use of linguistic information as effectively as possible in order to perform end-to-end MRC. We first present the general syntax-guided attentive architecture, building upon the recent advanced BERT[1] and then fit with task-specific layers for machine reading comprehension tasks.

[1]Note that our method is not limited to cooperate with BERT. We use it as the backbone because of the superior representation capacity and the strong baseline it provides.

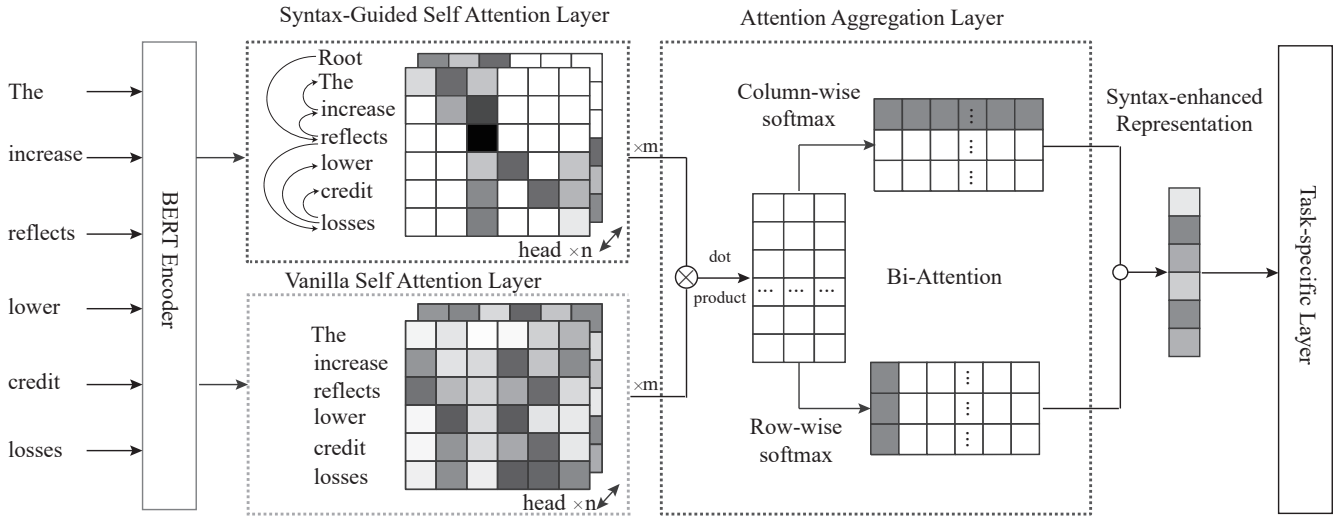


Figure 2: Overview of Syntax-Guided Network.

Figure 2 depicts the whole architecture of our model. The basis for our model is the fine-tuned BERT encoder introduced from (Devlin et al. 2018). We transform BERT embeddings into conditionally attended representation using our proposed syntax-guided self attention by only considering the dependency of interest (DOI) when calculating the weights for each token, which is intuitively a constrained and fine-grained interaction inside each sentence while the vanilla self attention is a kind of global processing by leveraging all the input tokens. Then we integrate the outputs from the global and conditional attention learning utilizing a bi-attention layer. At last, the resulting syntax-enhanced representation is passed to task-specific layers for final predictions.

BERT Encoder

Following the implementation of BERT (Devlin et al. 2018), the first token of every sequence is the special token [CLS] and the sequences are separated by the [SEP] token. The output of BERT H is then fed to our proposed syntax-guided attention layers to obtain the syntax-enhanced representation. We omit rather extensive formulations of BERT and recommend readers to get the details from (Devlin et al. 2018).

Syntax-Guided Network

Our syntax-guided representation is obtained by two steps[2]. Firstly, we pass the encoded representation to a syntax-guided self attention layer and a vanilla multi-head attention layer in parallel [3]. The corresponding outputs are then aggregated using a bi-attention network to form a syntax-enhanced representation for downstream tasks. Specifically, the syntax-guided self attention is designed to incorporate the syntactic tree structure information inside a multi-head attention mechanism to indicate the token relationships of each sentence which will be demonstrated as follows.

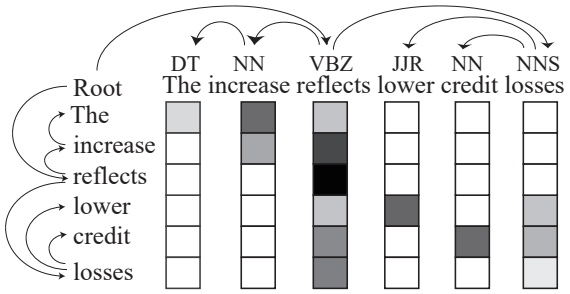


Figure 3: Syntax-guided self attention.

Syntax-Guided Self Attention Layer In this work, we first pre-train a syntactic dependency parser to annotate the dependency structures for each sentence which are then fed to SG-Net as guidance of token-aware attention. Details of the pre-training process of the parser are reported in Section .

To use the relationship between head word and dependent words provided by the syntactic dependency tree of sentence, we restrain the scope of attention only between word and all of its parent head words. In other word, we would like to have each word only attend to words of syntactic importance in a sentence, the parent head words in the view of the child word. As shown in Figure 3, instead of taking attention with each word in whole passage, the word *credit* only makes attention with its parent head words *reflects* and

[2]For brevity, we focus on a particular layer though we may use multiple layers which are stacked in the same way.

[3]Both vanilla and syntax-guided self-attention layers are extra parts after BERT 24-layer-encoder. We tried to apply DOIMASK to the 24-layer BERT but it only shows MARGINAL improvement. Thus we keep the whole 24-layer BERT encoder as it is and add two extra parallel parts followed by an aggregating part.

losses and itself in this sentence, which means that the DOI of *credit* contains *reflects*, *losses* along with itself.

To incorporate syntax, we introduce a syntax-guided multi-head self attention layer after the BERT encoder to attend to each token's syntactically related nodes, allowing the model to use the attention head as an oracle for syntactic dependencies.

Specifically, given input token sequence $S = (s_1, s_2, ..., s_n)$, we first use our parser to generate a sequence of index span of each token's subtree $SPAN_S = (span_1, span_2, ..., span_n)$, where $span_i = (j, k), 1 \leq i, j, k \leq n$. $span_i$ indicates the span of child nodes for head word s_i. To utilize the information provided by relationship between dependent token and parent tokens, we use $SPAN_S$ to build the DOI for each word using a dependency of interest mask DOIMASK of dimension $n \times n$:

$$\text{DOIMASK}[i, j] = \begin{cases} 1, & \text{if } j \in span_i \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Obviously, if DOIMASK$[i, j] = 1$, it means that token s_i is the parent node of token s_j.

We then project the last layer output H from the vanilla BERT into distinct key, value and query representations of dimensions $L \times d_k$, $L \times d_q$ and $L \times d_v$, respectively, denoted K_i', Q_i', V_i' for each head i. Then we perform a dot product to score key-query pairs with the dependency of interest mask to obtain attention weights of dimension $L \times L$, denoted A_i':

$$A_i{}' = \text{Softmax}\left(\frac{\text{DOIMASK} \cdot \left(Q_i{}' {K_i{}'}^T\right)}{\sqrt{d_k}}\right) \quad (2)$$

We then multiply attention weight A_i' by V_i' to obtain the syntax-guided token representations:

$$W_i' = A_i' V_i' \quad (3)$$

Then W_i' for all heads are concatenated and passed through a feed-forward layer followed by GeLU activations (Hendrycks and Gimpel 2016). After passing through another feed-forward layer, we apply a layer normalization to the sum of output and initial representation to obtain the final representations.

Attention Aggregation Layer To integrate syntax-guided token representations, we first feed the last layer of output of vanilla BERT H into a multi-head self attention layer (Vaswani et al. 2017) to obtain self-attended token representations $\tilde{H}$ and utilize a bi-attention network (Seo et al. 2016) to update representations with syntax-guided representations $\bar{H}$:

$$\begin{aligned} \tilde{H} &= \text{Self Attention}\,(H) \\ \bar{H} &= \text{Bi-Attention}\left(\tilde{H}, H'\right) \end{aligned} \quad (4)$$

Task-specific Adaptation

We focus on two types of reading comprehension tasks, *span-based* and *multi-choice* style which can be described as a tuple $< P, Q, A >$ or $< P, Q, C, A >$ respectively, where P is a passage (context), Q is a query over the contents of P, in which a span or choice C is the right answer A. For the span-based one, we implemented our model on SQuAD 2.0 task that contains unanswerable questions. Our system is supposed to not only predict the start and end position in the passage P and extract span as answer A but also return a null string when the question is unanswerable. For the multi-choice style, the model is implemented on RACE dataset which is requested to choose the right answer from a set of candidate ones according to given passage and question.

Here, we formulate our model for both of the two tasks and feed the output from the syntax-guided network to task layers according to specific task.

Given the passage P and a question Q and the choice C specially for RACE, we organize the input X for BERT as the following two sequences.

Span:	[CLS]	P	[SEP]	Q	[SEP]
Choice:	[CLS]	P || Q	[SEP]	C	[SEP]

where $||$ denotes concatenation.

The sequence is fed to BERT encoder mentioned above to obtain the contextualized representation *H* which is then passed it to our proposed syntax-guided self attention layer described in to obtain the syntax-enhanced representation $\bar{H}$. For span-based task, we fed $\bar{H}$ to Pointer Network and a no-answer (NA) Verifier. For multi-choice task, we fed it into the classifier to predict the choice label for the multi-choice model.

SQuAD 2.0 To enable fine-tuned BERT to competently work for SQuAD 2.0, we extend BERT by adding a *Pointer Network* and an *NA Verifier* for no-answer predictions. Our aim is a span of answer text and the prediction of the end position is dependent on the start position, thus we employ a Pointer Network (Vinyals, Fortunato, and Jaitly 2015) and fed $\bar{H}$ as the input to obtain the start and end probabilities, s and e, similar to the usage in various former MRC models such as R-Net (Wang et al. 2017):

$$s, e = \text{PointerNetwork}(\bar{H}) \quad (5)$$

For SQuAD 2.0 challenge, our model must not only answer questions when possible, but also determine when no answer is supported by the paragraph and abstain from answering, we also need a tactic to give no-answer predictions.

We pass s and e to softmax layer and employ batch matrix multiplication with $\bar{H}$ respectively to obtain weighted hidden representation. Meanwhile, we apply a first token pooling[4] on $\bar{H}$. Then we sum the above two weighted representations and concatenate with the pooled representation to fed

[4]The target answer for unanswerable question is set as the first token "[CLS]" during training, so we use the first token representation to make following no-answer prediction.

to a lineal layer to make the prediction that the question is answerable or not.

$$\begin{aligned} c_s &= \text{SoftMax}(s) \cdot \bar{H} \\ c_e &= \text{SoftMax}(e) \cdot \bar{H} \\ o_0 &= \text{FirstTokenPooling}(\bar{H}) \\ p_{na} &= \text{Linear}[(c_s + c_e) \circ o_0] \end{aligned} \tag{6}$$

The training objectives of our SQuAD model can be comprised by two parts, $\mathcal{L} = \alpha\mathcal{L}_{has} + \beta\mathcal{L}_{na}$, where $\mathcal{L}_{has}$ denotes the cross entropy loss for the start and end predictions and $\mathcal{L}_{na}$ is the binary cross entropy loss for no-answer prediction, α and β are hyper-parameter for weights of two loss functions.

$$\begin{aligned} \mathcal{L}_{has} &= y_s \log s + y_e \log e, \\ \mathcal{L}_{na} &= y_{na} \log p_{na} + (1 - y_{na}) \log(1 - p_{na}) \end{aligned} \tag{7}$$

where y_s and y_e are the targeted start and end positions, respectively. $y_{na} \in \{0, 1\}$ denotes the ground truth label of no-answer prediction.

For prediction, given output start and end probabilities s and e and no-answer probability p_{na}, we calculate the has-answer score $score_{has} = \max(s_k + e_l), 0 \leq k \leq l \leq n$ and the no-answer score $score_{na} = \lambda_1(s_0 + e_0) + \lambda_2 p_{na}$, where λ_1, λ_2 are coefficients. We obtain a difference score between has-answer score and the no-answer score as final score. A threshold is set to determine whether the question is answerable, which can be heuristically computed in linear time with dynamic programming. The model predicts the answer span that gives the has-answer score if the final score is above the threshold, and null string otherwise.

RACE As discussed in (Devlin et al. 2018), the pooled representation explicitly includes classification information during the pre-training stage of BERT, we expect the pooled to be overall representation of the input. Thus, the first token representation o_0 in $\bar{H}$ is picked out and passed to a feed-forward layer to give the prediction p. For each instance with n choice candidates, we update model parameters according to cross-entropy loss during training and choose the one with highest probability as the prediction when testing. The training objectives of our RACE model is defined as, $L(\theta) = -\frac{1}{N}\sum_i y_i \log p_i$ where p_i denotes the prediction, y_i is the target and i denotes the data index.

Experiments

Dataset and Setup

Span-based MRC As a widely used MRC benchmark dataset, SQuAD 2.0 (Rajpurkar, Jia, and Liang 2018) combines the 100,000 questions in SQuAD 1.1 (Rajpurkar et al. 2016) with over 50,000 new, unanswerable questions that are written adversarially by crowdworkers to look similar to answerable ones. For the SQuAD 2.0 challenge, systems must not only answer questions when possible, but also abstain from answering when no answer is supported by the paragraph. Two official metrics are selected to evaluate the model performance: Exact Match (EM) and a softer metric F1 score, which measures the weighted average of the precision and recall rate at a character level.

Multi-choice MRC Our Multi-choice MRC is evaluated on Large-scale ReAding Comprehension Dataset From Examinations (RACE) dataset (Lai et al. 2017), which consists of two subsets: RACE-M and RACE-H corresponding to middle school and high school difficulty levels. RACE contains 27,933 passages and 97,687 questions in total, which is recognized as one of the largest and most difficult datasets in multi-choice MRC. The official evaluation metric is accuracy.

Implementation

For the syntactic parser, we follow the deep biaffine attention based dependency parser from (Dozat and Manning 2016) using English dataset Penn Treebank (PTB) (Marcus, Santorini, and Marcinkiewicz 1993) to annotate our task datasets. We re-train the dependency parser by joint learning of constituent parsing (Kitaev and Klein 2018) using BERT as sole input which achieves very high accuracy: 97.00% UAS and 95.43% LAS on PTB test set[5]. Note this work is done in data preprocessing and our parser is not updated with the following MRC models.

After generating the dependency syntactic tree for each sentence, we use a simple depth-first search algorithm to calculate the left and right boundaries of each sub-tree as the index span of each node's children. Moreover, we convert the index span to attention mask which is convenient for range restriction of self attention layers.

For MRC model implementation, We follow the same fine-tuning procedure as BERT to avoid extra influence and focus on the intrinsic performance of our newly proposed method. We use the pre-trained cased BERT (whole word masking) as the backbone. We use Adam as our optimizer with an initial learning rate in {8e-6, 1e-5, 2e-5, 3e-5} with warm-up rate of 0.1 and L2 weight decay of 0.01. The batch size is selected in {16, 20, 32}. The maximum number of epochs is set to 3 or 10 depending on tasks. All the texts are tokenized using wordpieces, the maximum length is set to 384 for both of SQuAD and RACE. And the configuration for multi-head self attention is same as that for BERT. For SQuAD, coefficients are set as $\alpha = 0.5$, $\beta = 0.3$, $\lambda_1 = 0.5$. $\lambda_2 = 0.5$.

Main Results

To focus on the evaluation of syntactic advance and keep simplicity, we only compare with single models instead of ensemble ones.

Reading Comprehension Table 1 shows the result on SQuAD 2.0. Various state of the art models from the official leaderboard are also listed for reference. We can see that the performance of BERT is very strong. However, our model is more powerful, boosting the BERT baseline essentially with an increase of +3.1% EM and F1 on the test set. It also outperforms all the published works and achieves the comparative performance compared with more sophisticated mechanisms and pipeline systems.

[5]We report the results without punctuation of the labeled and unlabeled attachment scores (LAS, UAS)

Model	Test	
	EM	F1
Regular Track		
BiDAF-No-Answer (Rajpurkar, Jia, and Liang 2018)	59.2	62.1
DocQA (Rajpurkar, Jia, and Liang 2018)	59.3	62.3
DocQA + ELMo (Rajpurkar, Jia, and Liang 2018)	63.4	66.3
Joint SAN (Liu et al. 2018)	68.7	71.4
U-Net (Sun et al. 2018a)	69.2	72.6
RMR + ELMo + Verifier (Hu et al. 2018)	71.7	74.2
SLQA+†	71.5	74.4
BERT Track		
Human	86.8	89.5
BERT + DAE + AoA†	85.9	88.6
BERT + N-Gram Masking + Synthetic Self-Training†	85.2	87.7
BERT + ConvLSTM + MTL + Verifier†	84.9	88.2
SemBERT†	84.8	87.9
Insight-baseline-BERT†	84.8	87.6
BERT + MMFT + ADA†	83.0	85.9
$\text{BERT}_{\text{LARGE}}$	82.1	84.8
SG-Net	**85.2**	**87.9**

Table 1: Exact Match (EM) and F1 scores (%) on SQuAD 2.0 dataset for single models. Our model is in boldface. † refers to unpublished work. Besides published works, we also list competing systems on the SQuAD leaderboard.

Model	RACE-M	RACE-H	RACE
Human Performance			
Turkers	85.1	69.4	73.3
Ceiling	95.4	94.2	94.5
Leaderboard			
DCMN	77.6	70.1	72.3
$\text{BERT}_{\text{LARGE}}$	76.6	70.1	72.0
OCN	76.7	69.6	71.7
RSM	69.2	61.5	63.8
GPT	62.9	57.4	59.0
SG-Net	**78.8**	**72.2**	**74.2**

Table 2: Accuracy (%) on RACE test set for single models.

For RACE, we compare our model with the following latest baselines: Dual Co-Matching Network (DCMN) (Zhang et al. 2019), Option Comparison Network (OCN) (Ran et al. 2019) , Reading Strategies Model (RSM) (Sun et al. 2018b) and Generative Pre-Training (GPT) (Radford et al. 2018). Table 2 shows the result[6]. Turkers is the performance of Amazon Turkers on a random subset of the RACE test set. Ceiling is the percentage of unambiguous questions in the test set. From the comparison, we can observe that our model outperforms all baselines and achieves new state-of-the-art accuracy, which verifies the effectiveness of our proposed syntax enhancement.

[6]Our concatenation order of P and Q is slight different from the original BERT, which yields about 0.5%-1% accuracy improvement. Therefore, the result of our BERT baseline is higher than the public one on the leaderboard, thus our improved BERT implementation is used as the stronger baseline for our evaluation.

Model	EM	F1
Our model	85.1	87.9
-Syntax-Guided Network	84.1	86.8
Concatenation	84.5	87.6
Weighted Sum	84.8	87.7

Table 3: Performance of different aggregation methods on SQuAD 2.0 dev set.

Model Analysis

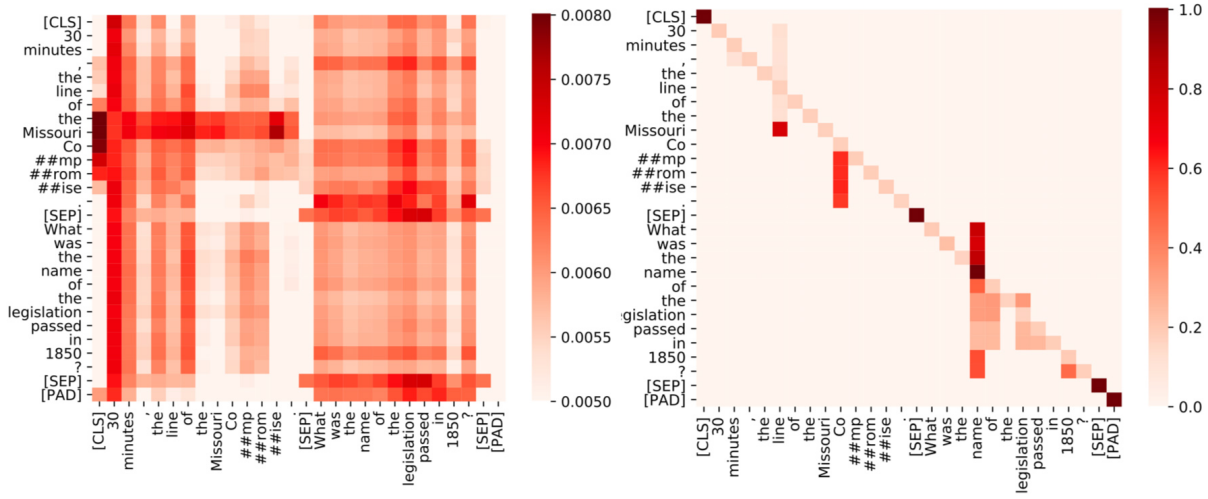
Syntax-Guided Attention

Table 3 shows a comparative study to evaluate the attention mechanism in our method. We observe that the performance drops dramatically without our proposed syntax-guided network. In SG-Net, we integrate the representations from syntax-guided attention layer and the vanilla self attention layer using a bi-attention. Actually, there are other simple operations for representation aggregation, such as *concatenation* and *weighted sum*, which are also involved in our comparison, which shows that using bi-attention is better for the final advance.

Visualization

To have an insight that how syntax-guided attention works, we draw attention distributions of the vanilla self attention and our proposed syntax-guided self attention[7], as shown in

[7]Since special symbols such as [PAD] and [CLS] are not considered in the dependency parsing tree, in our implementation, the child of these tokens is only confined to themselves. So in our syntax-guided layer, these special tokens will have value of 1 as weights over themselves and these weights do not matter as we will mask paddings in the following aggregation layer.



Passage:*...30 minutes, the line of the Missouri Compromise...* ***Question***:*What was the name of the legislation passed in 1850?* ***Answer***:*the Missouri Compromise*

Figure 4: Visualization of the vanilla self attention (left) and syntax-guided self attention (right). Weights of attention are selected from first head of the last attention layer. For the syntax-guided self attention, the columns with weights represent the DOI for each word in the row. For example, the DOI of *passed* contains {*name, of, legislation, passed*}. The weights are normalized by SoftMax to sum up to 1 for each row.

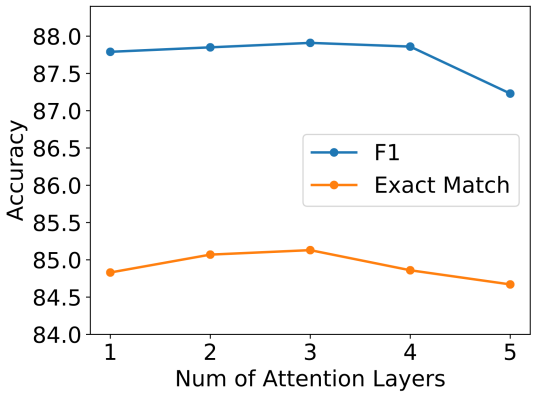


Figure 5: Case study of the number of attention layers on SQuAD 2.0 dev set.

Figure 4. With the guidance of syntax, the keywords *name*, *legislation* and *1850* in the question are highlighted, and *(the) Missouri*, and *Compromise* in the passage are also paid great attention, which is exactly the right answer. The visualization verifies our model is effective at selecting the vital parts, guiding the downstream layer to collect more relevant pieces to make predictions.

Attention Layers

Attention layer depth might be a potential factor that contributes to model performance. To investigate the influence, we conduct a case study on the number of syntax-guided self attention layers ranging from [1,5]. The comparison is in Figure 5. We observe that 3-layer attention shows to be the best and increasing the model depth would be harmful. This shows that a moderate depth of SG attention layer would be helpful.

Conclusions

This paper presents a novel syntax-guided framework for machine reading comprehension. Experiments on two major machine reading comprehension benchmarks involving span-based answer extraction (SQuAD 2.0) and multi-choice inference (RACE) show that our model can yield new state-of-the-art or comparative results in both extremely challenging tasks. This work empirically discloses the effectiveness of syntactic structural information for text modeling. The proposed attention mechanism also verifies the practicability of using linguistic information to guide attention learning and can be easily adapted with other tree-structured annotations.

References

[Bowman et al. 2016] Bowman, S. R.; Gauthier, J.; Rastogi, A.; Gupta, R.; Manning, C. D.; and Potts, C. 2016. A fast unified model for parsing and sentence understanding. *arXiv preprint arXiv:1603.06021*.

[Chen and Manning 2014] Chen, D., and Manning, C. 2014. A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 740–750.

[Cui et al. 2017] Cui, Y.; Chen, Z.; Wei, S.; Wang, S.; Liu, T.; and Hu, G. 2017. Attention-over-attention neural networks for reading comprehension. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)* 1832–1846.

[Devlin et al. 2018] Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Dhingra et al. 2017] Dhingra, B.; Liu, H.; Yang, Z.; Cohen, W. W.; and Salakhutdinov, R. 2017. Gated-attention readers for text comprehension. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)* 1832–1846.

[Dozat and Manning 2016] Dozat, T., and Manning, C. D. 2016. Deep biaffine attention for neural dependency parsing. *ICLR*.

[He et al. 2017] He, L.; Lee, K.; Lewis, M.; Zettlemoyer, L.; He, L.; Lee, K.; Lewis, M.; Zettlemoyer, L.; He, L.; and Lee, K. 2017. Deep semantic role labeling: What works and whats next. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)* 473–483.

[Hendrycks and Gimpel 2016] Hendrycks, D., and Gimpel, K. 2016. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *ICLR*.

[Hermann et al. 2015] Hermann, K. M.; Kocisky, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; and Blunsom, P. 2015. Teaching machines to read and comprehend. *Advances in Neural Information Processing Systems (NIPS 2015)* 1693–1701.

[Hill et al. 2015] Hill, F.; Bordes, A.; Chopra, S.; and Weston, J. 2015. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*.

[Hu et al. 2018] Hu, M.; Peng, Y.; Huang, Z.; Yang, N.; Zhou, M.; et al. 2018. Read+ verify: Machine reading comprehension with unanswerable questions. *arXiv preprint arXiv:1808.05759*.

[Joshi et al. 2017] Joshi, M.; Choi, E.; Weld, D. S.; and Zettlemoyer, L. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)* 1601–1611.

[Kadlec et al. 2016] Kadlec, R.; Schmid, M.; Bajgar, O.; and Kleindienst, J. 2016. Text understanding with the attention sum reader network. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)* 908–918.

[Kitaev and Klein 2018] Kitaev, N., and Klein, D. 2018. Constituency Parsing with a Self-Attentive Encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2676–2686.

[Lai et al. 2017] Lai, G.; Xie, Q.; Liu, H.; Yang, Y.; and Hovy, E. 2017. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 785–794.

[Liu et al. 2018] Liu, X.; Li, W.; Fang, Y.; Kim, A.; Duh, K.; and Gao, J. 2018. Stochastic answer networks for squad 2.0. *arXiv preprint arXiv:1809.09194*.

[Ma et al. 2018] Ma, X.; Hu, Z.; Liu, J.; Peng, N.; Neubig, G.; and Hovy, E. 2018. Stack-Pointer Networks for Dependency Parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 1403–1414.

[Marcheggiani and Titov 2017] Marcheggiani, D., and Titov, I. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. *arXiv preprint arXiv:1703.04826*.

[Marcus, Santorini, and Marcinkiewicz 1993] Marcus, M. P.; Santorini, B.; and Marcinkiewicz, M. A. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics* 19(2).

[Mudrakarta et al. 2018] Mudrakarta, P. K.; Taly, A.; Sundararajan, M.; and Dhamdhere, K. 2018. Did the model understand the question? *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)* 1896–1906.

[Nguyen et al. 2016] Nguyen, T.; Rosenberg, M.; Song, X.; Gao, J.; Tiwary, S.; Majumder, R.; and Deng, L. 2016. Ms marco: A human generated machine reading comprehension dataset. *ArXiv:1611.09268v2*.

[Peters et al. 2018] Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. *The 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018)* 2227–2237.

[Radford et al. 2018] Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training. *Technical report*.

[Rajpurkar et al. 2016] Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100,000+ questions for machine comprehension of text. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)* 2383–2392.

[Rajpurkar, Jia, and Liang 2018] Rajpurkar, P.; Jia, R.; and Liang, P. 2018. Know what you don't know: Unanswerable questions for SQuAD. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)* 784–789.

[Ran et al. 2019] Ran, Q.; Li, P.; Hu, W.; and Zhou, J. 2019. Option comparison network for multiple-choice reading comprehension. *arXiv preprint arXiv:1903.03033*.

[Roth and Lapata 2016] Roth, M., and Lapata, M. 2016. Neural semantic role labeling with dependency path embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 1192–1202.

[Seo et al. 2016] Seo, M.; Kembhavi, A.; Farhadi, A.; and Hajishirzi, H. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

[Strubell et al. 2018] Strubell, E.; Verga, P.; Andor, D.; Weiss, D.; and McCallum, A. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of*

the 2018 Conference on Empirical Methods in Natural Language Processing, 5027–5038.

[Sun et al. 2018a] Sun, F.; Li, L.; Qiu, X.; and Liu, Y. 2018a. U-net: Machine reading comprehension with unanswerable questions. *arXiv preprint arXiv:1810.06638*.

[Sun et al. 2018b] Sun, K.; Yu, D.; Yu, D.; and Cardie, C. 2018b. Improving machine reading comprehension with general reading strategies. *arXiv preprint arXiv:1810.13441*.

[Vaswani et al. 2017] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

[Vinyals, Fortunato, and Jaitly 2015] Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer networks. In *Advances in Neural Information Processing Systems (NIPS 2015)*, 2692–2700.

[Wang et al. 2017] Wang, W.; Yang, N.; Wei, F.; Chang, B.; and Zhou, M. 2017. Gated self-matching networks for reading comprehension and question answering. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)* 189–198.

[Wang et al. 2018] Wang, Y.; Liu, K.; Liu, J.; He, W.; Lyu, Y.; Wu, H.; Li, S.; and Wang, H. 2018. Multi-passage machine reading comprehension with cross-passage answer verification. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)* 1918–1927.

[Wang, Yan, and Wu 2018] Wang, W.; Yan, M.; and Wu, C. 2018. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)* 1705–1714.

[Wang, Zhang, and Zong 2017] Wang, S.; Zhang, J.; and Zong, C. 2017. Learning sentence representation with guidance of human attention. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017)*, 4137–4143.

[Yu, Lee, and Le 2017] Yu, A. W.; Lee, H.; and Le, Q. 2017. Learning to skim text. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, 1880–1890.

[Zhang and Zhao 2018] Zhang, Z., and Zhao, H. 2018. One-shot learning for question-answering in gaokao history challenge. *COLING*.

[Zhang et al. 2018] Zhang, Z.; Li, J.; Zhu, P.; and Zhao, H. 2018. Modeling multi-turn conversation with deep utterance aggregation. *COLING*.

[Zhang et al. 2019] Zhang, S.; Zhao, H.; Wu, Y.; Zhang, Z.; Zhou, X.; and Zhou, X. 2019. Dual co-matching network for multi-choice reading comprehension. *arXiv preprint arXiv:1901.09381*.

[Zhang, Huang, and Zhao 2018] Zhang, Z.; Huang, Y.; and Zhao, H. 2018. Subword-augmented embedding for cloze reading comprehension. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*, 1802–1814.